AS-Path Prepending: there is no rose without a thorn

Pedro Marcos*
FURG
pbmarcos@furg.br

Lars Prehn*
MPI for Informatics
lprehn@mpi-inf.mpg.de

Lucas Leal
UFRGS
lsleal@inf.ufrgs.br

Alberto Dainotti
CAIDA, UC San Diego
alberto@caida.org

Anja Feldmann
MPI for Informatics
anja@mpi-inf.mpg.de

Marinho Barcellos
University of Waikato
marinho.barcellos@waikato.ac.nz

ABSTRACT

Inbound traffic engineering (ITE)—the process of announcing routes to, e.g., maximize revenue or minimize congestion—is an essential task for Autonomous Systems (ASes). AS Path Prepending (ASPP) is an easy to use and well-known ITE technique that routing manuals show as one of the first alternatives to influence other ASes' routing decisions. We observe that origin ASes currently prepend more than 25% of all IPv4 prefixes.

ASPP consists of inflating the BGP AS path. Since the length of the AS path is the second tie-breaker in the BGP best path selection, ASPP can steer traffic to other routes. Despite being simple and easy to use, the appreciation of ASPP among operators and researchers is diverse. Some have questioned its need, effectiveness, and predictability, as well as voiced security concerns. Motivated by these mixed views, we revisit ASPP. Our longitudinal study shows that ASes widely deploy ASPP, and its utilization has slightly increased despite public statements against it. We surprisingly spot roughly 6k ASes originating at least one prefix with prepends that achieve no ITE goal. With active measurements, we show that ASPP effectiveness as an ITE tool depends on the AS location and the number of available upstreams; that ASPP security implications are practical; identify that more than 18% of the prepended prefixes contain unnecessary prepends that achieve no apparent goal other than amplifying existing routing security risks. We validate our findings in interviews with 20 network operators.

CCS CONCEPTS

• **Networks → Network measurement**.

ACM Reference Format:
Pedro Marcos, Lars Prehn, Lucas Leal, Alberto Dainotti, Anja Feldmann, and Marinho Barcellos. 2020. AS-Path Prepending: there is no rose without a thorn. In *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3419394.3423642

*Both authors have contributed equally to the paper.

IMC '20, October 27–29, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8138-3/20/10…$15.00
https://doi.org/10.1145/3419394.3423642

1 INTRODUCTION

Many Internet Autonomous Systems (ASes) receive significantly more traffic than they send. They often use inbound traffic engineering (ITE) to influence the link through which they receive traffic based on economic considerations (e.g., transit cost) or operational demands (e.g., latency, packet loss, capacity). ITE has become even more important, as there are more options for inter-AS connectivity due to, e.g., IXPs (Internet eXchange Points), PNIs (Private Network Interconnects), and an overall increase of peering [9, 58, 71, 74, 75]. Border Gateway Protocol (BGP)-enabled ITE techniques include AS-Path Prepending (ASPP) [15, 22, 76], selective or more-specific prefix announcements [27], BGP communities [23, 63], or Multi Exit Discriminator (MED) values [25, 41].

In this paper, we focus on understanding ASPP deployment and the potential issues associated with it. ASPP is a straightforward, easy-to-use technique that is often mentioned among the first ITE techniques by router vendors [19, 21, 26, 35, 43]. It is a technique where an AS artificially inflates the BGP AS path by inserting (subsequent) duplicate entries of its ASN. Since the length of an AS path is the second most important tie-breaker in BGP best path selection, ASPP may steer traffic from one route to another. However, its effect depends on route propagation and the routing decisions made by other ASes. Despite (or because of) its simplicity and its inherent limitations, the appreciation of ASPP among operators and researchers is mixed. On the one hand, ASPP—unlike other ITE techniques—does not need any support from other ASes, nor deaggregatable prefixes. On the other hand, its need, effectiveness, and predictability have been questioned [37, 50, 65]. In addition, there have been concerns about the extent to which ASPP can amplify existing routing insecurities [38, 39, 64], and reports of improper ASPP configurations triggering bugs in router software [79, 80].

Motivated by the mixed views about the ASPP method, we investigate the current use of ASPP and find that more than 30% of ASes use it. Thus, to contribute to an informed discussion, we address three fundamental questions:

(i) **How do ASes use prepending?** To put effectiveness and risk into context, we first identify and characterize the *policies* ASes apply (i.e., the number of prepends used for each prefix) when using ASPP. Even when using data from all route collectors over the last decade, limited route visibility [16, 29, 47] poses a significant challenge. We deal with it by conducting interviews with more than 20 operators and by cross-checking our results with private data sources from large Internet players.

(ii) **How effective is prepending?** Among both operators and academics, the opinion on whether ASPP is effective as an ITE

technique diverges and often depends on the position of an AS in the routing ecosystem. For example, Quoitin et al. [50] showed that ASPP is unpredictable using their vantage point. We claim that the effectiveness of ASPP is indeed diverse—it depends on the vantage point within the routing system and the number of available upstreams. We highlight this behavior by actively testing a large number of vantage points and varying the number of upstreams.
(iii) **Does prepending amplify existing routing security risks?** Often, a "malicious" route needs to be the shortest path in order to be adopted. ASPP facilitates the spreading of malicious routes by making the legitimate paths longer. While one may observe malicious routes in public BGP data, the lack of suitable what-if scenarios (i.e., how would the scenario change with a larger prepend size) poses a significant challenge. We shed light on this topic by systematically emulating numerous prefix hijacks from many vantage points.

We approach these questions using both active and passive measurements. We use passively collected routing information from Isolario [34], RIPE RIS [1], and RouteViews [2] to perform a longitudinal study. We then use the PEERING testbed [57, 59] to systematically explore ASPP from a large number of vantage points and emulate many scenarios through targeted BGP route announcements and probing traffic.

We summarize our main contributions as follows:

- We perform a longitudinal characterization of ASPP utilization and identify that, despite the community mixed opinions, its utilization has been steadily increasing. We find that, on May 2020, 30% of the ASes prepend at least one of their prefixes, resulting in 25% of the IPv4 prefixes being originated with ASPP (see § 4).
- We also identify that ASes mainly originate their prefixes with two distinct prepending sizes (e.g., without prepend and with two extra prepends) to indicate their preference for inbound traffic. Surprisingly, we also find that roughly 6k ASes originate a total of more than 28k prefixes with a single prepending size (different than zero), thus resulting in no ITE effect (see § 5).
- We discover that in scenarios with only two upstreams, ASPP effectiveness is strongly dependent on the vantage point. Yet, when using many upstreams, ASPP shifts traffic from most incoming sources (see § 6).
- Using active experiments, we identify that prefixes with three prepends are highly suitable for prefix hijacking. Today, ASes originate more than 15k prefixes with at least three prepends, increasing the risks of widespread route leaks or prefix hijacking with no apparent ITE benefit (see § 7).

We discuss ethical considerations in Appendix § A, and to foster reproducibility and research on ASPP, we make all of our analysis code available to the research community.[1]

2 PRIMER ON PATH PREPENDING

ASPP is an ITE technique in which an AS adds its own AS number n extra times ($n \geq 1$) before originating/propagating a BGP route, thus artificially increasing the resulting AS-Path length by n. We refer to n as the *prepend size*. Whenever an AS receives a route announcement, it chooses the best path according to a list of tie-breaking rules. The first rule relies on local preference. To affect the route selection of remote ASes [52], an AS uses ASPP to inflate the AS Path to influence the second tie-breaking rule: to prefer the shortest AS path. (If the tie persists, route origin and MED values are among the remaining tie-breakers.)

In Figure 1a, we illustrate the use of ASPP by an AS with two neighbors. AS A announces a prefix P to both neighbors with different prepend sizes. By making one path longer, AS A attempts to influence remote ASes to send traffic through AS B. The success of this attempt will depend on how remote ASes will receive the announcements. In Figure 1b, we depict a case where ASPP can influence the decision of AS F. Even though the path traversing AS C has fewer ASes than the one going through AS B, AS F prefers the second path as it is the shortest. In Figure 1c, we show a case where the ASPP by AS A cannot influence the decision of AS F as it has fewer prepends—AS F prefers the path traversing AS C as it has the smallest AS path length. These cases underline that ASPP cannot guarantee remote route changes and the resulting ingress traffic distribution.

We distinguish two forms of prepending. If the AS prepending is the originator, we refer to it as *origin-prepending*; otherwise, we refer to it as *intermediate-prepending*. When an AS prepends on behalf of another AS, we refer to this particular form of intermediate-prepending as *remote-prepending*. In such cases, ASes can use BGP communities or web interfaces to ask the other ASes to prepend. ASes use remote-prepending to affect path choices that are beyond the reach of origin-prepending.

ASes can use ASPP for *load balancing* among upstreams, to *minimize transit cost* (by moving traffic away from an expensive upstream), or to establish *backup* links. Among the reasons mentioned by operators for ASPP popularity are its ease of use on commercial routers, its efficiency in steering incoming traffic, and the requirements and shortcomings of alternate mechanisms.

3 DATASETS AND DATA SANITATION

To analyze ASPP utilization, we rely on (BGP) MRT data publicly available from Isolario[2] [34], RIPE RIS[1], and Route Views [2]. We use the following datasets in our analyses.

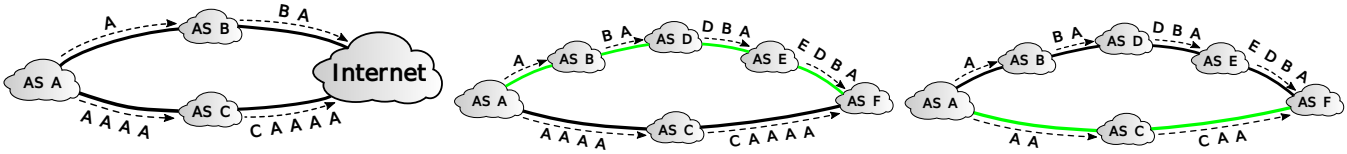
$\text{BGP}_{\text{Continuous}}$: This dataset contains RIB snapshots from all available BGP collectors on March 1st, 2020 at 0:00 UTC+0. In addition, it contains all subsequent update files until April 1st, 2020, at 0:00 UTC+0. If an update file is missing in a collector's repository, we add the next available RIB snapshot to capture potentially missed changes.

$\text{BGP}_{\text{Weekly}}$: This dataset contains data for each Monday between January 1st, 2018, and May 4th, 2020. For each day, we use the RIB snapshots from all available BGP collectors at 0:00 UTC+0 and all consecutive updates for that day. We, again, compensate for missing files.

$\text{BGP}_{\text{Monthly}}$: This dataset contains data for the 15th day of every month between January 15th, 2010, and April 15th, 2020. We generate the data of a single day in the same way as for the previous dataset.

[1] https://gitlab.mpi-klsb.mpg.de/lprehn/imc20_aspp

[2] Isolario was hit by lightning on July 30th, 2019, leading to some missing files until August 16th, 2019 —we find that the impact to our analysis is minimal.



(a) AS A announces the prefix *P* with ASPP to one of its neighbors.

(b) ASPP in effect.

(c) Route adoption regardless of ASPP.

Figure 1: AS-Path Prepending behavior.

ROAS: Rather than using tools (such as Routinator [45]) to preprocess RPKI data, we take advantage of the preprocessed data provided by Chung et al. [18]. We use data for the same days as in the BGP_{Weekly} dataset.

RIR: This dataset contains the (extended—if available) delegation files from AFRINIC [3], APNIC [5], ARIN [6], LACNIC [36], and RIPENCC [54] for all days in the $BGP_{Monthly}$ dataset.

Data sanitation. Before analyzing our BGP data, we remove well-known artifacts. First, we remove bogon routes, i.e., routes that lead only to reserved address space [66] or routes that contain ASes currently reserved by IANA [33]. Similarly, we remove all routes to prefixes less specific than /8. This step ensures that we only analyze default-free routing information.[3] We further remove all routes for which the path contains a loop. The sanitation, up to this point, removed ~3.36M (0.7 %) routes and reduced the number of prefixes from ~1.29M to ~932k (-28 %) using the last snapshot of the BGP_{Weekly} dataset as reference (we find similar values for other snapshots). To avoid making false inferences due to lack of visibility, we only analyze prefixes visible by at least one-third of the BGP monitors set on the corresponding date. When analyzing how many monitors see each prefix, we find a clear separation between locally and globally visible prefixes regardless of the exact year (in Appendix § B we report more details). Notably, the last step reduced the number of unique prefixes to ~803k.

4 TRENDS IN THE USE OF ASPP

Previously reported metrics about ASPP differ across studies, with the most recent results being from 2016 [11, 25, 27, 68]. To understand ASPP utilization better, we analyze its trends over the last decade using the $BGP_{Monthly}$ dataset. We note our numbers represent lower bounds of the actual ASPP utilization, as *(i)* the visibility of route collectors is limited [16, 29, 47]; *(ii)* prepended paths tend to be less attractive than non-prepended ones; *(iii)* we sanitize our data (see § 3).

One-third of all ASes use origin-prepending. Figure 2 shows the fraction of ASes using ASPP (for IPv4) separated by prepending type (recall § 2): *origin-prepending* or *intermediate-prepending*. First, we see that the fraction of ASes using ASPP has increased slightly, from ~28% (9.4k) on January 15th, 2010 to ~31.4% (21.6k) on April 15th, 2020, with most ASes using origin-prepending. Similarly, we observe a small increase in intermediate prepending—from 4.7% (1.6k) on January 15th, 2010, to 5.5% (3.8k) on April 15th, 2020. [4] We also see a very small fraction (<1%) using only intermediate prepending, some of which might be due to ASes offering remote-prepending, e.g., via BGP communities.

The fraction of prepended prefix-origin pairs and addresses has increased slightly. Next, we focus on prefixes. We consider a prefix/IP address as prepended if at least one AS has added its ASN more than once (consecutively) to the path. In Figure 3 we observe that the increase of prefixes with origin-prepending is similar to the one observed respective to ASes—from ~21.3% (65.2k) on January 15th, 2010 to ~25.9% (207.7k) on April 15th, 2020.[5] Regarding the intermediately prepended prefixes, we observe that for the entire $BGP_{Monthly}$ dataset, (almost) all prefixes contain prepend in all snapshots. Such a condition happens because there are transit ASes (especially Tier-1s) that prepend most prefixes before redistributing them to at least one of their neighbors.

For IP addresses we see a larger increase of origin-prepending—from ~26.2% (570 million) in January 15th, 2010 to ~38.9% (1.1 billion) on April 15th, 2020. This more pronounced increase is likely correlated to the exhaustion of the IPv4 address space and the fact that prefixes more specific than /24 tend to propagate less [49].

To not overestimate the numbers, we check how many of these addresses are covered by more-specific non-prepended prefixes.[6] We find that on April 15th, 2020, only 9.3% of the origin-prepended address space was reachable through more-specific non-prepended announcements. We also observe that for 79% of the cases, the less specific announcements were visible in more monitors than the more-specific ones, indicating that 9.3% may be an over-estimate.

Discussion. Despite various public call-outs of the drawbacks of ASPP [37–39, 64], we observe that its use has not decreased. Most operators were surprised by the results. According to them, the long-term use of ASPP is a sign of either bad capacity planning or inexperienced network engineers. Nevertheless, some argued that many factors might render ASPP more attractive than other ITE techniques for network operators, including not being able to obtain an address space larger than a /24 in certain regions (e.g., RIPE[55, 56]); the capital required to expand the infrastructure; the simplicity of ASPP; and its prominence in router vendor handbooks.

Focus on origin-prepending and IPv4. As the large number of intermediately prepended prefixes is the result of the routing policies of a small number of large ASes (e.g., Tier-1s), for the remainder of this paper, we focus on the (far more common) origin-based prepending. Also, we choose to focus on IPv4 prepending, as

[3] As opposed to cases in which an AS uses the default route (i.e., 0.0.0.0/0) to send traffic to some/all destinations.

[4] The spike on the fraction of ASes applying intermediate prepending corresponds to the period in which a set of experiments [63] involving the use of BGP communities to manipulate ASPP was taking place.

[5] We also analyzed ASPP growth considering only monitors available on January 15th, 2010 and find similar behavior.

[6] Recall BGP longest-prefix-matching prefers routes for (non-prepended) more-specific prefixes over (prepended) less-specific ones.

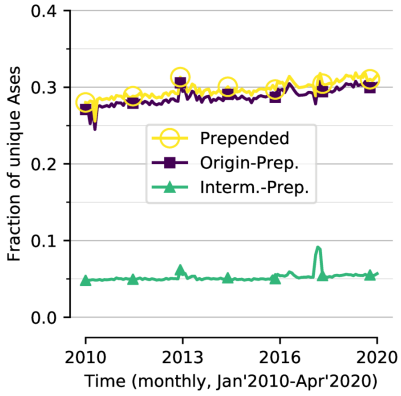


Figure 2: Fraction of ASes deploying ASPP.

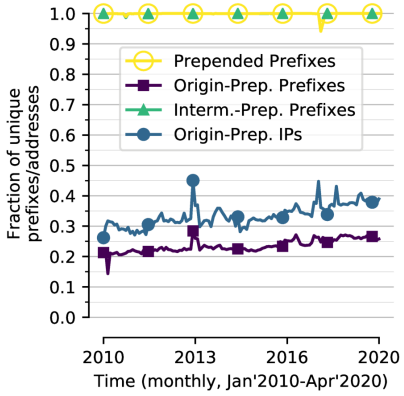


Figure 3: Fraction of Prefixes/IPs with ASPP.

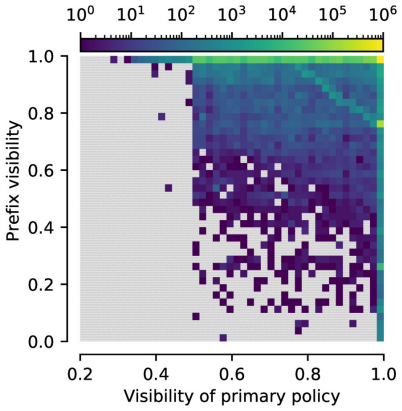


Figure 4: Prefix-origin primary policy consistency across a month.

IPv6 accounts for only 6% of the total cases of prepending on April 15th, 2020.

5 PREPENDING POLICIES IN THE WILD

To understand how operators use ASPP with the prefixes they originate, we identify different *policies* and look at their prevalence in-the-wild, both in terms of prefix-origin pairs (§ 5.1) and ASes (§ 5.2). In our analyses, we find a surprising incidence of a seemingly innocuous form of ASPP, called *uniform* prepending, which we thus investigate more closely (§ 5.3). Last, we examine the evolution of prepending sizes in different geographic service regions (§ 5.4). As in the previous section, we consider only prefix-origin pairs visible by at least one-third of all BGP monitors.

5.1 ASPP policies: Prefix-origin pairs

We identify four different prepending policies that can be used in a prefix-origin pair. They are *(i) no-prepend*: no visible prepended route; *(ii) uniform*: the only visible prepend size is N, where $N > 0$; *(iii) binary*: visible routes either have prepend size M or N, where $M, N \geq 0$ and $M \neq N$; *(iv) diverse*: the number of different prepend sizes in the visible routes exceeds two.

ASes tend to stick with a (per-prefix) policy over time. Our first focus is on policy *consistency*—how often does an AS change a prefix prepending policy? For this analysis, we use the $\text{BGP}_{\text{Continuous}}$ dataset, in which we identify roughly 2.3 million unique prefix-origin pairs. For each pair, we define as its *primary policy* the one we observe more often throughout the full month (among *no-prepend*, *uniform*, *binary* and *diverse*). We examine the stability of the primary policy for a prefix-origin pair with respect to its visibility period. Figure 4 shows a heatmap, where colors indicate the number of pairs in each cell. We observe a concentration in the top right section of the plot, which corresponds to 54% of prefix-origin pairs, indicating that they are visible all the time and never change their primary policy. We repeated the analysis for another month (Sep. 2019) and found similar primary policy stability, which allows us to adopt weekly ($\text{BGP}_{\text{Weekly}}$) or monthly ($\text{BGP}_{\text{Monthly}}$) snapshots without any loss in the subsequent analyses.

The use of policies has been stable, with *binary* policy being more common. Using the $\text{BGP}_{\text{Weekly}}$ dataset, we examine the use of prepending policies for prefix-origin pairs between January 1st, 2018, and May 4th, 2020. In Figure 5, we see that the most common prepending policy is *binary*, followed by *diverse* and *uniform* policies. Their popularity remains largely stable, if considering the proportion to the full set of pairs: *diverse* increased from 4.5% (30k) to 6.1% (50k), *binary* decreased from 17.2% (114k) to 15.9% (131k), while *uniform* remained at 3.6% (24.4k to 29.4k)[7]. We note that the trend regarding the use of more fine-grained policies might be related to the increasing connectivity level of ASes (e.g., connecting to more IXPs). For the sake of comparison, we looked at the use of *uniform* prepending back in January 2010, and it was 2.7% (8.2k). The consistent presence of *uniform* policy through time is surprising since, in theory, it should not influence any remote BGP decisions. We take a closer look at this phenomenon in § 5.3.

More prepending during COVID-19 lockdown. We also note that between February and April of 2020, the number of prefix-origin pairs with ASPP reached approximately 30% (4% increase). Such a peak is likely related to the lockdown measures due to COVID-19, which resulted in people staying more time at home [28, 72, 73]. In this period there have been reports of traffic increases [4, 20, 44], which also resulted in content providers such as Netflix and Youtube stopping streaming in 4k to save bandwidth [31].

We believe that the higher use of ASPP during this period was necessary for network operators to handle the increasing demands of traffic while upgrading their links (as ASPP use has decreased in May). When we discussed this with network operators, some of them mentioned that they were observing more use of ASPP, especially during large live events streamed on Youtube [42], and that some of their transit customers were requesting capacity upgrades.

[7] In May 4th, 2020, all types of prepending combined represented 25.6% of all prefix-origin pairs, corroborating findings in § 4.

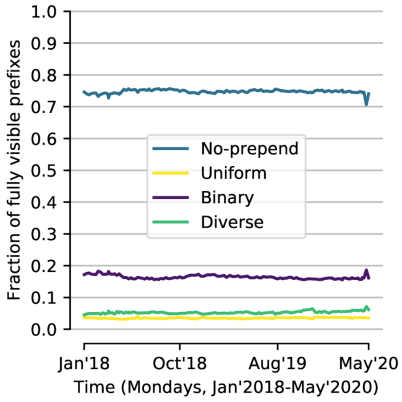


Figure 5: Prefix-origin: Fractions through time of visible prefixes per ASPP policy.

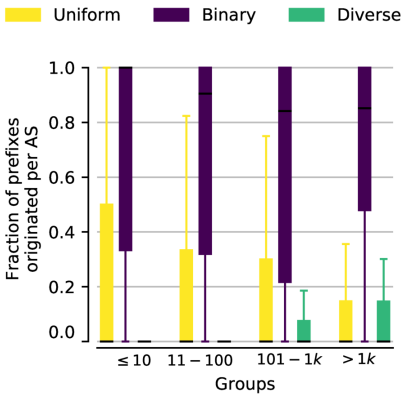


Figure 6: *Mixed* policy ASes grouped by # of prefixes.

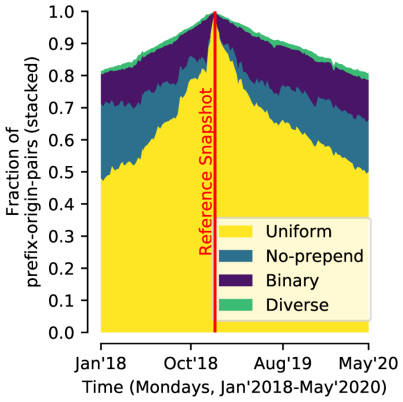


Figure 7: Fractions of prepending policies through time for a fixed set of uniform-prepend prefixes.

5.2 ASPP policies: ASes

We now change the perspective of our policy analysis to ASes. We differentiate per-AS ASPP policies as follows. When an AS employs a single policy for all prefixes it originates, we say it adopts one of the four policies already defined: *no-prepend*, *uniform*, *binary* or *diverse*. Otherwise, we say an AS employs a *mixed* set of policies.

Most ASes that prepend use multiple policies. Using the BGP_{Weekly} dataset, we analyze the use of AS prepending policies between January 1st, 2018, and May 4th, 2020. We observe that more than 30.8% (20.8k) of the ASes prepend at least one prefix they originate (consistently with § 4), and most ASes use *mixed* prepending policies on May 4th, 2020. Among those using a single policy, the most common case is the *binary*policy, followed by *uniform* and *diverse*, respectively. Over time the fractions of different policies are substantially stable, with only a slight increase in all but *binary* policies. (In May 4th, 2020 we observe the following percentages: *uniform* 2.5%, *diverse* 1.4%, *mixed* 16.4%, and *binary* 10.4%). Once again, we note an increase in fine-grained prepending policies, which may be associated with a general increase in AS connectivity. We consider conceptually more straightforward for an AS to employ a single policy. Nevertheless, ITE may require the AS to use *mixed* policies, such as *binary* for some prefixes and *no-prepend* for others.

ASes with *mixed* policies mainly use *binary* policies. Next, we focus on ASes using a *mixed* set of policies and analyze the fraction of prefixes using each of the prepending policies. We group these few ASes according to the number of originated prefixes, in four "bins": $1-10$, $11-100$, $101-1000$, 1000+ prefixes. For each AS in a bin, we calculate the fraction of prefixes for each policy and present it as a boxplot in Figure 6 for May 4th, 2020 (we observe similar behavior for other snapshots). The plot shows only ASes that employ a *mixed* set of policies[8], and we observe that for these ASes, the most common is the *binary* policy (in all bins). We also find, confirming our intuition, that the fraction of the *diverse* policy increases with the number of prefixes an AS originates (more pronounced for the two larger bins). Conversely, we see the fraction of uniformly prepended prefixes decreasing (with AS size).

[8]The same plot for different dates, namely all snapshots of the BGP_{Weekly} dataset showed similar results.

5.3 Uniform prepending

Uniform prepending is widespread. There is no apparent reason for an AS to use the same prepending size for all its neighbors when originating a prefix, as it implies no differentiation among them. Nevertheless, on May 4th, 2020, we observe more than 29k (3.6%) uniformly prepended prefix-origin pairs originated by 5.8k (8%) ASes, out of which 1.7k (2.5%) ASes prepend all their prefixes uniformly.

Some prefixes use the uniform policy consistently. The use of *uniform* policy might be the result of temporary events. To determine whether this is a common case, we pick all the (25.8k) uniformly-prepended prefix-origin pairs on a specific date (December 31st, 2018), and use the BGP_{Weekly} dataset to show the fractions of policy type change for these prefixes in the preceding/following months. Figure 7 shows that the total number of prefixes decreases both sides, as up to 20% prefixes were not visible earlier or stop being visible afterward. We observe that both before and after December 31st, 2018 the fractions of *no-prepend*, *binary*, and *diverse* (for this fixed set of prefix-origin pairs) increase while *uniform* decreases. In other words, for some of these prefixes, *uniform* prepending was temporary. On the other hand, for the entire period, we see at least 50% of prefix-origin pairs using the *uniform* policy. Since there is no guarantee that these are the same prefixes, we look into it further.

Between January 1st, 2018, and May 4th, 2020, we observe 1.16M prefix-origin pairs in our BGP_{Weekly} dataset. Out of these, 108k prefixes are uniformly prepended in at least one snapshot, and 3.4k (originated by 1.1k ASes) use this policy the entire time—henceforth referred to as consistently uniform. We also note that another 13.1k (originated by 4.3k ASes) are uniformly prepended for at least one year, *continuously*. Thus, counter-intuitively, we find that a substantial number of ASes, roughly 6% on the Internet, are making consistent use of *uniform* prepending.

Uniform prefix prepending is dominated by small ASes. How large are those interesting cases of ASes uniformly prepending all their prefixes? To answer this question, we determine the total number of prefixes each of these ASes originate. Taking May 4th, 2020, as an example, there were 848 (out of 1717) ASes with only a single prefix. Another 767 ASes originated between 2 and 10 prefixes, and 89 ASes, between 11 and 50. The remaining 13 ASes originated more than 50 prefixes, *all* of them uniformly prepended, with the largest one originating 379 prefixes. Among the larger ASes (with 50+ prefixes), we identified a large online social network, two universities, and several ISPs. These ASes are from North America, South America, and Asia.

We then check for how long these ASes used the *uniform* policy. We find that 6k ASes (out of 74k that we observe when combining all snapshots) uniformly prepend all prefixes in at least one snapshot, and 263 ASes used this policy between January 1st, 2018, and May 4th, 2020. We also see that other 716 ASes uniformly prepended all their prefixes for at least one year. From the group of 13 ASes that on May 4th, 2020, were uniformly prepending all of their 50+ prefixes, we find the following: one AS used it at least since January 1st, 2018, five for at least the past two years, one for the past 22 months, three for at least one year. The others consecutively prepended between 2 and 5 months.

We account for potential artifacts when measuring *uniform* prepending. Even though our sanitation ensures global visibility of all prefixes, missing interconnections may cause prefixes to incorrectly appear as *uniformly* prepended. There might be additional private network interconnects and peering links that are not visible to the BGP monitoring infrastructure [16, 47, 75]. We use two different approaches for cross-checking the results. First, we use bdrmapIT [40], a state-of-the-art tool, to infer interconnections based on public traceroutes from CAIDA's Archipelago (Ark) [12] between March 25th, 2020 and April 4th, 2020. We picked Ark traceroutes as it contains measurements to each /24 sub prefix from multiple vantage points. We then compare the list of interconnections from bdrmapIT with the ones we observe in our snapshot from March 30th, 2020 (the mid-point of our traceroutes). On our reference date (March 30th, 2020), 5.8k ASes were originating at least one prefix uniformly prepended. With bdrmapIT, we identify additional interconnection links for 1.7k (29%) of these ASes. Nevertheless, for the other 71% ASes originating uniformly prepended prefixes, bdrmapIT did not add any additional links. For the 263 ASes that uniformly prepended all their prefixes in all snapshots of the BGP_{Weekly} dataset, we identify new links for only 18 of them. We note that even though we identify new links, we cannot draw any inference regarding the BGP announcements made through those links.

The second cross-check is to increase our visibility into the BGP routing system with data from two large global CDNs (each connected to more than 200 peering infrastructures) and one regional CDN present in more than 25 peering infrastructures. We choose CDNs since they have many private peering interconnections and need excellent visibility within the routing system for their operations. When checking their private data for all prefixes uniformly prepended in all snapshots of the BGP_{Weekly} dataset, we observe more diverse policies for only 51 of those prefixes. Thus, we can conclude that our inferences are valid for the vast majority of the uniform cases.

Some of these prefixes carry large volumes of traffic. Some operators mentioned that consistently uniformly prepended prefixes might only carry little traffic, reducing the need to care about them. To check this hypothesis, we use a large European IXP as our vantage point on April 28th, 2020. We check the traffic volumes to and from each of the consistently uniformly prepended prefixes and observe that some of them carry as much traffic as prefixes of large social networks.[9]

To provide a picture of the traffic associated with all consistently uniformly prepended prefixes in our vantage point, Figure 8 shows the fraction of bytes flowing towards each prefix (as well as in both directions) relative to the prefix with the most significant amount of traffic. For 57% of the prefixes, we do not observe any traffic towards them, and for 35%, we observe traffic from them, but not towards them. We note that only a few prefixes (<2%) carry representative volumes of traffic, either considering one or both directions. The vast majority of the prefixes we observe carry small volumes of traffic. While we cannot guarantee that other vantage points would observe similar numbers, we can conclude that, contrary to network operators' intuition, some of the consistently uniformly prepended prefixes carry substantial traffic volumes.

Many plausible causes for *uniform* prepending. Is there any *practical* explanation for the use of *uniform* prepending? We investigate this aspect by interviewing network operators, and report here a summary of potential causes: *Loss of a neighbor:* an AS may have used ASPP to differentiate between multiple upstreams but later terminated the relationship with some. Indeed, we observe that many (77% on May 4th, 2020) of the uniformly prepended prefixes are propagated via a single neighbor. *Lack of knowledge:* A reoccurring opinion is that many network operators, especially from small ASes, have limited understanding of BGP. Indeed, our analysis showed that many of the cases of *uniform* prepending were from small ASes. *Procrastination for stability:* Some network operators know about the presence of ASPP but are reluctant to remove it, out of fear of negatively affecting their reachability and/or routing stability in general. *Good news travels fast—bad news, slowly:* Some operators indicated that *uniform* prepending may help implement ITE policies when needed quickly. Instead of waiting to insert prepends when some change is needed, an AS can prepend in advance, and when the time comes, remove from one upstream to indicate a preferred route. Since "good news" travel fast, such an approach provides faster BGP convergence. *Sibling artifacts:* One operator pointed out that there might be cases in which two or more sibling ASes originate the same prefix, but with different prepending policies. We analyze this possibility using the CAIDA AS2Org dataset [14] and the data from May 4th, 2020. We find 17 cases in which two or more sibling ASes individually announced the same prefix, one uniformly prepended and the other with a different policy, resulting in a non-uniform policy. Strikingly, in 16 out of 17 cases, one of the ASes announces using *uniform* and the other one with a different policy. In one case, both ASes originate the prefix uniformly prepended, but with different prepending sizes. *Other ASes ignoring prepends:* One operator argued that *uniform*

[9] We are not allowed to disclose the actual byte counts of each prefix.

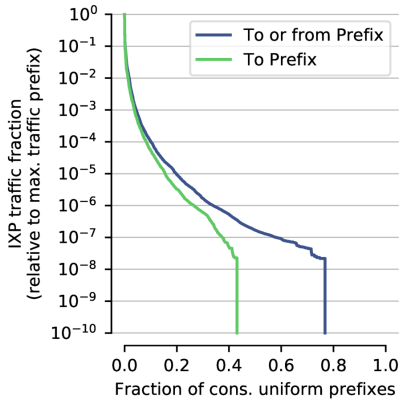


Figure 8: Uniform prefix-origin IXP traffic on April 28, 2020.

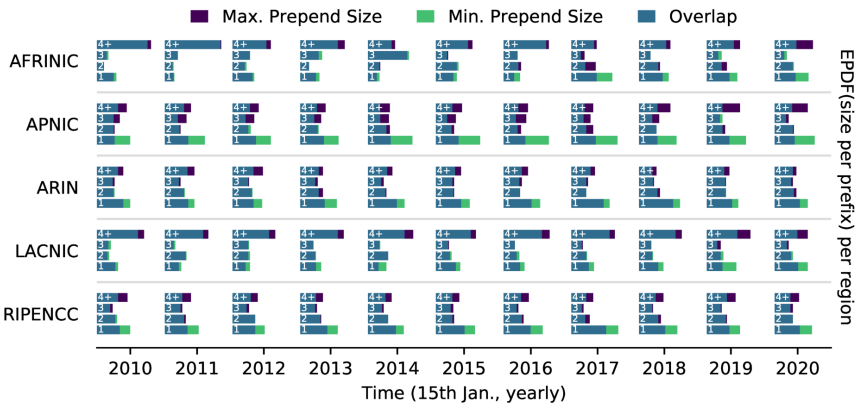


Figure 9: Prefix-origin: Prepend size by region across time.

prepending might even lead to the desired traffic shift due to route-optimizers ignoring all prepends on one upstream and not on the others.

Looking at two relevant cases of *uniform* prepending. To validate our observations and to understand some of the actual reasons why ASes uniformly prepend their prefixes, we reached out to network operators from two ASes that have been originating uniformly prepended prefixes for more than one year.[10] One is a regional ISP that uniformly prepends 25 prefixes (out of 100+), while the other is a large online social network uniformly prepending all its 80+ prefixes. The operators from the regional ISP confirmed that the *uniform* prepending was unintentional and attributed it to legacy configurations and changes to their upstreams. The large online social network also confirmed that they were using *uniform* prepending unintentionally: the prepends are a result of how their internal routing platform operates. Since then, none of these ASes have removed the *uniform* prepends.

5.4 Prepending sizes

We use the $\text{BGP}_{\text{Monthly}}$ dataset to track if ASes changed the *number of prepends* they use over time. Since different service regions have distinct characteristics (e.g., availability of peering infrastructures [69]), we analyze them individually. We use the delegation files from the Routing Information Registries (RIRs) to identify the prefix region. While we acknowledge that there might be some misclassification, e.g., for global ASes, transferred prefixes, or due to IPv4 address delegations, we expect it to provide valid data for most prefixes. For each prefix, we analyze its minimum (non-zero) and maximum prepend size, i.e., if an ASes originates a prefix with 0, 2, and 3 prepends, its minimum prepended size is 2, and its maximum is 3. Figure 9 shows the results as a set of subplots, one for each service region and year. Each subplot shows a histogram for both the minimum (green) and maximum (purple) prepend sizes across all prepended prefixes. The blue bars represent the overlap between the green and the purple bars.

Prepending sizes are polarized and consistent among regions. We observe that the prepending size distributions for ARIN and RIPE, which hardly change during the decade, are *polarized*: most prefixes either have a prepending size of one or at least four.[11] LACNIC and AFRINIC are different: in 2010, there is no polarization, with a substantial number of prefixes with at least four prepends, while in 2020 polarization happens with a more significant incidence of prepending of size one. The change happens gradually over time, but in AFRINIC, the period 2014–2017 was an exception: prepending sizes varied "rapidly" and somewhat unpredictably. Towards 2020, the observable differences between the service regions become negligible—they are all polarized. In APNIC, the span between max and min prepend sizes increased, indicating more polarization, with an even more fine-grained set of prepending policies.

When we discussed these results with operators, they pointed out that the Internet infrastructure changed significantly throughout the decade, particularly for LACNIC and AFRINIC. Before 2015, many routes within Africa took long inter-continental detours [30]. In order to use intra-continental paths whenever possible, ASes resorted to excessive prepending. With the increased availability of IXPs and peering within each region, intra-continental path diversity increased [24]. This may have reduced the need for excessive prepending, thus reducing prepend sizes.

6 EVALUATING ASPP EFFECTIVENESS

Given the widespread use of ASPP, in this section we explore the propagation of prepended routes and how effective ASPP is today.

Prepended paths propagate less than non-prepended ones. The common assumption is that the larger the prepend size of a route is, the less a network operator will expect it to propagate. Thus, prepending should mainly affect routing in the local neighborhood of an AS. To investigate how prepended prefixes propagate,

[10] We note that not all ASes are interested in discussing aspects of their operational practices. While discussing with network operators might not be enough for generalization, their comments allow us to provide insights regarding uniformly prepended prefixes.

[11] In July 2019, we spot an AS originating four prefixes with 905 prepends, which is the maximum number of prepends we observe in our datasets.

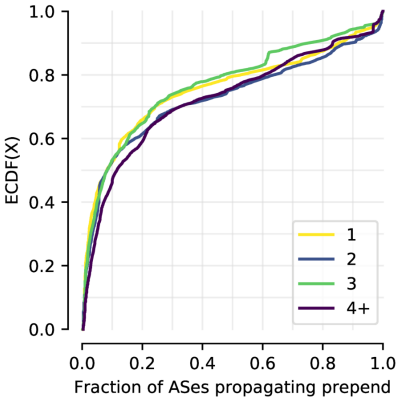


Figure 10: Fraction of ASes adopting longer alternative.

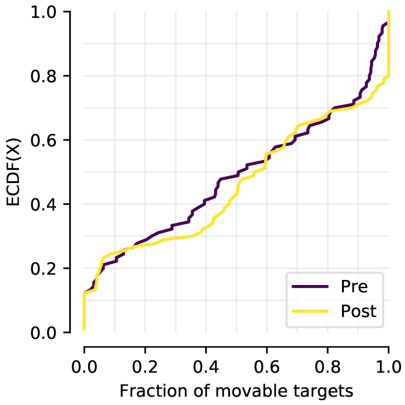


Figure 11: Fraction of potentially movable targets.

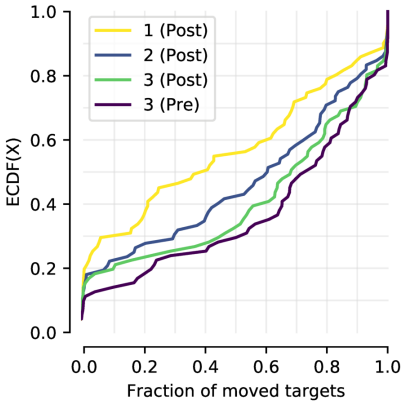


Figure 12: Fraction of actually moved targets.

we analyze all prefixes with a binary prepending policy where the prefix originator has not prepended one of the alternatives. For each prefix, we compute the fraction of ASes (out of those that we observe propagating the prefix) that propagate each alternative. Figure 10 shows the results for May 4th, 2020 (we observe a similar behavior for other snapshots). We observe that in 70% of the analyzed cases, independently from the prepend size, the prepended alternative traverses fewer ASes than the non-prepended one. While it may seem that the prepend size has no direct effect on route propagation, a more plausible explanation is that the ASes are tuning their prepend size to control how far the prefixes can propagate. Figure 10 shows that the distributions of the intended scopes of propagation are quite similar for different prepending sizes.

Nevertheless, it is unclear to which extent the adoption of a prepended path impacts the actual traffic flow, since (a) different routers in an AS may pick different preferred paths, (b) BGP monitors cover only a subset of ASes, and (c) some ASes might even remove ASPP (see § 7.1). Thus, we run active measurement experiments using the PEERING testbed as our vantage point. The PEERING testbed offers unique possibilities for our experiments. First, it operates on a geographically diverse set of locations—we refer to each location as Point of Presence (PoP). Second, each PoP has a diverse set of upstreams—the number of upstreams and the degree of connectivity of the individual upstreams differ among PoPs. Third, the PEERING testbed allows us to originate probing traffic towards a diverse set of targets using ICMP, TCP, and UDP.

On an abstract level, we create a scenario where we announce a route with prepends for some upstreams and no prepends (preferred) for others. Then, we use ICMP/TCP/UDP ping probes towards a diverse set of targets to generate response traffic towards the PEERING testbed AS. If the traffic enters via one of the preferred ASes, we refer to the result as a "hit", otherwise as a "miss". We note that the PEERING testbed allows us to correctly identify in which of the POPs the response has arrived.

Target selection. We base our target selection on Rapid 7's list of HTTP/1.1 GET responses [51]. We first select only IP addresses that responded with the HTTP status code "200 OK" when queried by an HTTP/1.1 GET request. To sample a diverse set of targets, we first map IPs to ASes by performing a longest prefix match on the closest snapshot of our $\text{BGP}_{\text{Weekly}}$ data set. Afterward, we classify ASes as follows: *(i)* we use a public list [70] to identify Tier-1 ASes; *(ii)* we use CAIDA's AS type classification [13] to identify "Content" and "Enterprise" ASes; *(iii)* we identify the remaining ASes as either "Access" or "Transit"—based on whether we observe them *only* as origin ASes in the $\text{BGP}_{\text{Weekly}}$ snapshot[12]. Since the Tier-1 class only contains 23 ASes, we use all of them as target ASes. For each of the remaining classes, we sampled 250 target ASes, resulting in 1023 targeted ASes. By running our own GET requests, we make sure to select only ASes for which 20 different IPs respond, resulting in a final target set of 20460 IP addresses.

Upstream selection. While the PEERING testbed has hundreds of upstreams, only roughly 20 provide transit. Since ASPP will have no effect if the prefix is subject to prefix aggregation [52] by a remote AS, we check how "well" our prefix propagates. We then announce it in one upstream per time and check how many monitors observe the prefix without aggregation. We filter out those upstreams that propagate our prefix to less than 200 monitors after 30 minutes of convergence. After this step, 11 transit providers—present at 10 different PoPs—remain. For the sake of simplicity, we focus on only one transit provider per PoP. We use the following PoPs: Amsterdam (A), Clemson University (C), Georgia Institute of Technology (GA), GRnet (GR), Northeastern University (N), Seattle (S), UFMG (UF), Utah (UT), University of Washington (UW), and University of Wisconsin (W).

Experiments. Each experiment employs a pair of PoPs, and we repeat it for all combinations and for different sizes of prepending (none, one, two, and three). We then run three sets of experiments. In the first set, we pick one upstream from each PoP and announce our test prefix on both—one with prepending and one without prepending. In the second set, we announce the prefix to all upstreams, prepending for all but one. In the last set of experiments, we announce the prefix to all upstreams but prepend to only one.

[12] All those ASes are in the "Access/Transit" class in CAIDA's classification.

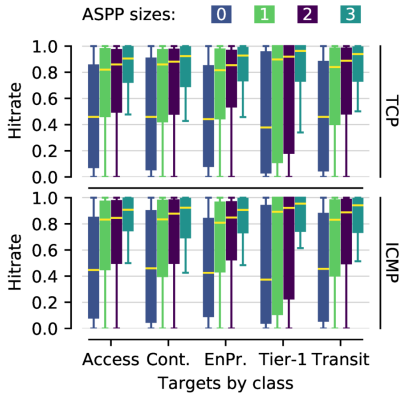


Figure 13: Hitrates by protocol and target class.

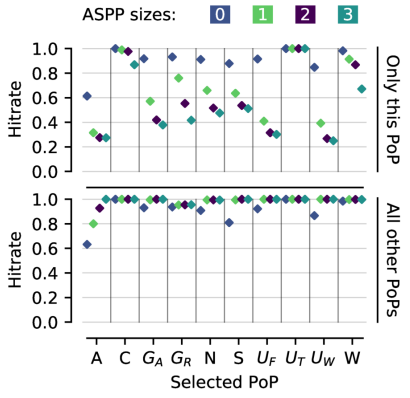


Figure 14: Hitrates when prepending 1 (top)|N-1 (bottom) PoPs.

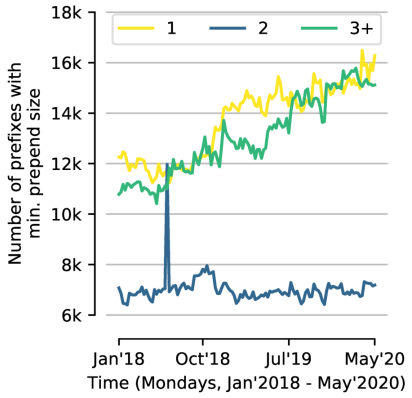


Figure 15: Prefix-origin: Pairs with at least X prepends.

We refer to specific choices of prepending size, upstreams, and experiment-class as an *iteration*. Our experiments took place between August 27th, 2020, and September 21st, 2020.

Iteration schedules. We deploy two similar iteration schedules that only differ in their first two rounds of announcements. For the "*Post*"-schedule, we start each iteration announcing our prefix P via all upstreams without any prepending. After waiting 15 minutes to allow BGP to converge, we announce P with X prepends via the chosen upstream (Appendix § C shows a detailed graphical timeline). For the "*Pre*"-schedule, we do the opposite: we first announce P with X prepends via the chosen upstream; we wait for 15 minutes, and finally, announce P without prepending via all but the chosen upstream. We employ both schedules to contain the impact of route age as a tie-breaking factor. We wait another 15 minutes for both schedules for BGP convergence before starting a 25 minutes long probing period. Each probe consists of ICMP, TCP, and UDP pings since the transport protocol can potentially bias the forwarding path [7, 48]. We probe once per minute all targets. To reduce probing bursts, we spread the packets evenly across the one minute time interval. To enable targets to opt-out, we embed our contact information in the payload of every probe. The cleanup phase starts 30 minutes after the start of the probing phase. Thus, we have a 5-minute break to ensure that the last responses can arrive before we withdraw the prefix. To allow for BGP to converge and minimize the risk of BGP Route Flap Damping, we wait for 30 minutes before starting a new iteration.

Data cleaning. In our results, we only consider those targets for which we see a significant number of responses: we require at least 10 of 25 probes for each protocol to be successful. However, we notice multiple probing artifacts, including many duplicates, additional ICMP packets, and RST packets. Thus, we first clean our data in the following manner: *(i)* we remove duplicate packets by relying on ICMP and TCP sequence numbers—since we sent SYN-packets, we receive duplicate TCP SYN-ACKs and RESET packets caused by receiver timeouts; *(ii)* we only consider ECHO-REPLY ICMP packets—we remove, in particular, ICMP TYPE 3 (destination port unreachable) for UDP and TCP probes; *(iii)* we hardly get any responses to the UDP probes, hence, we do not further consider them; *(iv)* for a given iteration, we remove all targets for which we receive responses via multiple interfaces—this can, e.g., occur if an AS uses load balancing. Overall, these steps remove less than 3% of the unique iteration-target combinations for ICMP and TCP.

Location matters when using only two upstreams. First, we look at how different prepending sizes influence routing behavior when using only two upstreams. Figure 11 shows the ECDF for the fraction of potentially movable targets (i.e., those targets initially routed via the later prepended upstream) per iteration and iteration type. We observe that our tested upstream-pairs cover the entire spectrum of scenarios, i.e., few, medium, and many potentially movable targets. Given this insight, we investigate how many of the potentially movable targets have been moved by each prepend size. Figure 12 shows an ECDF for the fraction of actually moved targets (based on the number of potentially movable targets) per PoP combination. We observe that the effectiveness of prepending can strongly depend on the location (for around 20% of cases, ASPP has moved no targets, while for another 20% , it moved almost all targets). We further observe that the change from a prepend of size one to a prepend of size two has a much larger impact than the change from size two to three. While we observed that the *Pre*-schedule performs slightly better than the *Post*-schedule (see the effectiveness of the maximum prepend size for both schedule types in the figure), the route age did not significantly affect our results. When manually looking into our data, we observe that for some pairs, the traffic shifts can happen either way (e.g., GRnet and Northeastern University), whereas for others, prepending has little effect (e.g., for Georgia Institute of Technology and Clemson University). The lack of effectiveness of ASPP might be caused by the low connectivity degree of the ASes. However, we observe a different result for Northeastern University despite the same number of upstream providers of Clemson University and Georgia Institute of Technology. This highlights that location (not only connectivity) plays an essential role in the effectiveness of ASPP. In addition, we observe that traffic shifts, in most cases, are not

gradual; instead, there is a minimum prepend size necessary to shift a majority of the targets.

Effectiveness differs based on the target class. Based on the above results, we study if the probing protocol and target class change the effectiveness of ASPP. Figure 13 shows a box plot of per-target hit rates (i.e., fraction of experiments where the target was a hit) per prepend size, network type, and transport protocol. Comparing the top plot with the one at the bottom, highlights that the overall hit rates are the same for both protocols. Comparing the different network classes, shows that Tier-1 targets were the hardest to influence using ASPP; however, the difference between target classes is not statistically significant.

With many upstreams, ASPP is able to shift almost all targets consistently. Finally, we analyze prepending's effectiveness for more than two upstreams (second and third sets of experiments). Figure 14 shows the hit rate per PoP when only one PoP is prepended (*top*) or when all other PoPs are prepended (*bottom*). In the experiments in which all but one upstream use prepending (bottom plot), we observe that, except for few cases, even small prepending sizes steer all traffic to the non-prepended upstream. The same holds for the inverse (top plot). If only a single PoP prepends, its hit rate quickly drops with increasing prepend size; however, it never drops to zero.

Discussion. In conclusion, with only two upstreams, the effectiveness of ASPP is strongly dependent on the location within the routing ecosystem; whereas with many upstreams, ASPP is able to shift almost all targets consistently. This notion is consistent with our conversations with operators. On the one hand, a few operators told us that certain ASes (mostly CDNs) might ignore prepends during their best-route selection, leading to limited effectiveness. On the other hand, many operators claimed that prepending works well for their networks most of the time, highlighting that ASPP is indeed useful for certain ASes.

7 SECURITY IMPLICATIONS

In this section, we shed light on some of the security concerns of ASPP that the community recently brought to network operators' attention [37–39, 64]. We first analyze if ASes manipulate prepended paths, i.e., remove prepends. Then, we experimentally verify and evaluate—on the Internet—the potential impact of hijacking of prepended prefixes as a basis for discussing the increased vulnerability of prepended prefixes. Finally, we estimate if ASes that prepend their prefixes also use RPKI-based Route Origin Validation (ROV) to protect their prefixes against hijacks.

7.1 Is removing prepends a common case?

When propagating routes, ASes should prepend their ASN at least once and keep the remaining AS path unchanged [52]. Nevertheless, no mechanism prevents an AS from modifying the path. Indeed, there have been reports about ASes (possibly) removing prepends from paths [77]. An AS might remove (all) prepends from a path to create a shorter path and potentially attract more traffic. Besides malicious behavior (i.e., for traffic inspection), potential reasons include economics (e.g., to earn revenue by trying to increase the 95th-percentile of the exchanged traffic [46, 60]) and performance (e.g., to adapt traffic flow).

Consider the scenario of Figure 1b, where AS A announces the prefix P to its two upstreams (*AS B* and *AS C*). *AS B* receives the non-prepended route, while *AS C* receives a route with three extra prepends. AS A would expect that most of the traffic towards prefix P would arrive on the link with *AS B*. Now suppose that *AS C* intends to increase its revenue. If *AS C* removes (all) prepends added by AS A, it makes its route shorter and more attractive to others.

Methodology. We check if we can observe such behavior happening systematically in the wild. We perform active measurements, since using passive BGP data to infer path manipulations is difficult (e.g., due to lack of visibility). Using the PEERING testbed, between May 3rd, 2020 and May 12th, 2020, we announce our prefix with three prepends via one of the PEERING's upstreams, and 30 minutes later, we withdraw it. After the withdrawal, we wait for another 30 minutes before starting a new iteration using a different upstream. After iterating through all available upstreams, we analyze all BGP updates (visible at route collectors) for our prefix. If we identify an update where at least one prepend is missing, we mark the upstream for further analysis. In the end, we do an in-depth experiment for the marked upstreams (that removed at least one prepend). For each of these, we announce a prepended path and wait 15 minutes for BGP to converge. Then we manually inspect the chosen best routes via BGP looking glasses and route servers to identify which AS is likely the one that is removing prepends. Then, we withdraw the prefix. After 45 minutes, we check the next marked upstream. We announce our prefix using 231 different upstreams, resulting in more than 22k observed paths and 738 traversed ASes.

Prepending removal is rare. After manual investigation, we find that a single AS removed prepends, on a single path (in a previous run of this experiment, in September 2019, we found three ASes consistently removing prepends). We cannot attribute this to malicious behavior, as we learned from conversations with network operators that some route optimizers might remove prepends.

7.2 Can ASPP "ease" prefix hijackings?

By artificially increasing the AS path length, an AS makes a route "less attractive" to other ASes. However, this behavior may create opportunities for other ASes to hijack this prefix for a larger part of the Internet ecosystem, since longer paths are more suitable for prefix hijacking [8]. Recall the scenario of Figure 1b. Let us assume an AS X (un)intentionally originates a path for prefix P that contains AS A as the first hop. ASes that use a prepended path are more likely to adopt this new route (originated by AS X) since it is shorter than the one originally propagated by *AS C*. Possible variations of this scenario reflect different prefix hijacking types (e.g., using an illegitimate origin, or manipulating the path so that the malicious AS is next to the actual origin AS [17, 61]) and route leaks [32, 62]. In all these scenarios, a "bad" route may replace a legitimate prepended route.

Routes with at least three prepends are more vulnerable to prefix hijacking. Recall that there have been reports that ASPP may increase the risk of prefix hijacking [37, 38, 64]. To better understand to which extent different lengths of ASPP facilitate the adoption of hijacked routes, we performed an experiment using the PEERING testbed.

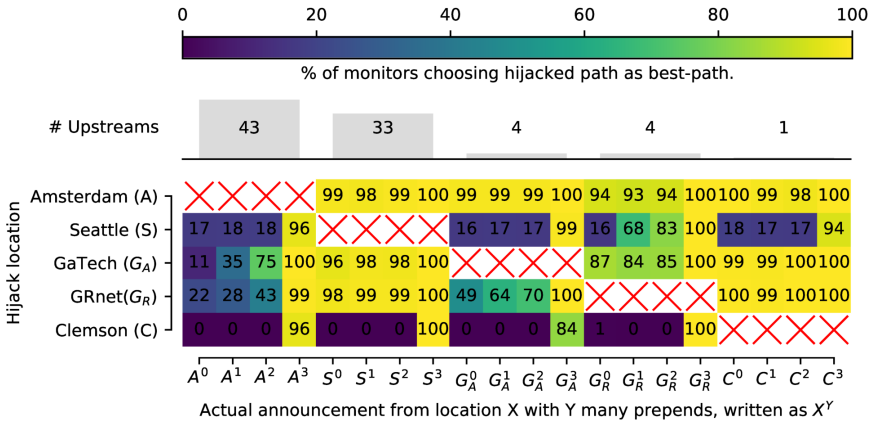


Figure 16: Hijacking: Fraction of BGP monitors adopting a hijacked route.

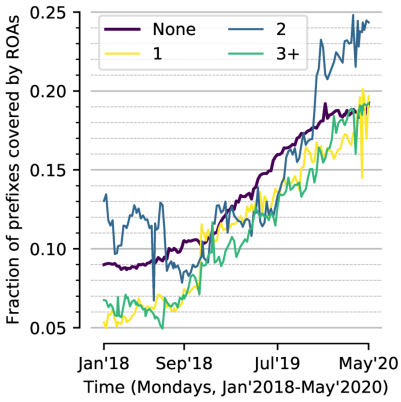


Figure 17: Fraction of prepended prefixes with ROAS.

We ran our measurements between January 13th, 2020, and January 17th, 2020. In each round, we announce our prefix using two different ASNs as originators. We first announce it via one of the PEERING's PoPs to all attached upstreams using AS61574 as originator and 0, 1, 2, or 3 prepends. Then, 15 minutes later, we also announce the same prefix via a second PEERING PoP to all its upstreams without prepends using AS61575.[13] 30 minutes after the second announcement, we withdraw all routes for the prefix. 30 minutes later, we repeat the experiment using a different combination of PoPs and/or number of prepends. We select PoPs based on their location and number of upstreams: Amsterdam, 44; Seattle, 33; GaTech, 4; GRnet, 4; and Clemson, 1. To capture the prefix hijack's impact, we analyze the fraction of BGP monitors that adopted the "hijacked route" via AS61575.

Figure 16 shows the fraction of monitors that adopted the hijacked route per pair of PoPs and prepend size. The results confirm the intuition that the likelihood of prefix hijacking succeeding increases with the number of prepends. Overall, we find that if the initial announcement used three prepends, at least 94% of the monitors adopted the hijacked route, even when the hijacking location only has single upstream (e.g., Clemson). Still, connectivity plays a vital role in the success of prefix hijacking. For all cases where we attempted to hijack the prefix from Amsterdam (a highly connected PoP), we succeeded for at least 93% of the monitors. In contrast, when we hijack via Clemson (a poorly connected PoP), we only succeed if the other PoP is prepending three times. Except for Clemson, all other PoPs were able to hijack Seattle mostly. Also, Seattle (with 33 upstreams) had less success in hijacking routes unless they had three prepends, which highlights the complexities of the Internet routing ecosystem.

Our results using a uniform prepending policy are an indication of how ASPP can increase the success of a hijacking attempt. While in § 5 we show that ASes uniformly prepend many prefixes, most ASes use a binary or diverse prepending policy, whereby one route is often not prepended. This means that the increased risk of hijacking only applies to the part of the Internet that chooses the prepended route.

More than 18% of prepended prefixes include apparently unnecessary prepending, which increases their exposure to hijacking and route leaks.

While for most prepended prefixes (169k) the minimum prepend size is 0, still many ASes originate prefixes with at least one prepend to all their neighbors, which can increase their exposure to hijacks and/or route leaks. For example, on May 4th, 2020, 6.9k ASes originated 38.5k prefixes with this characteristic (18.6% of all prepended prefixes). Among these, 29.4/7.4/2k used a uniform, binary, diverse policy. All these routes contain at least one unnecessary prepend—all their policies can be implemented with less prepending (at least as observable at the BGP monitors).[14] To further understand such potential risks, we use the $\text{BGP}_{\text{Weekly}}$ dataset to analyze the *minimum prepending size* for all prepended prefixes. Based on results in Figure 15, we see that the above finding holds across time, and also that the number of affected prefixes has grown.

7.3 RPKI-covered prepended prefixes

One of the main techniques for enhancing routing security is RPKI (Resource Public Key Infrastructure). RPKI allows ASes to create ROAs (Route Origination Authorizations) for each of their prefixes that other ASes can use to validate routes using ROV (Route Origin Validation) [18, 53, 67]. Although ROV cannot avoid the removal of prepends (see § 7.1), it can protect against prefix hijacking attacks in which the hijacker alters the origin AS [17, 61]. Given that ASPP potentially increases the exposure during hijacking attacks, we analyze to which degree prepended prefixes are protected by ROAs.

Most prepended prefixes are not covered by ROAs. We use the ROAS dataset to check which of the prepended prefixes in the $\text{BGP}_{\text{Weekly}}$ dataset has a ROA object. Figure 17 shows the coverage by ROAs of all prefixes in which all alternatives contain prepend and for those prefixes without prepend (none).

[13] The PEERING testbed requires us to add AS47065 after the originating AS to the AS path.

[14] We confirmed this conclusion in our conversations with network operators.

We observe first that the fraction of prefixes covered by ROAs has been increasing in the past years. On the other hand, we note that no more than 25% of the prefixes in each prepending class have ROAs[15]. This indicates that most prepended prefixes are not even partially protected against prefix hijacking attacks, regardless of the minimum number of prepends.

Discussion. Our security related results confirmed the assumptions that most network operators shared with us. Nevertheless, some of them argued that coming close to a specific traffic distribution may be more important to some ASes than reducing the potential impact of prefix hijacks—especially with the added security due to the increasing ROV deployment.

8 RELATED WORK

Previous studies already focused on characterizing ASPP, understanding its effectiveness, and pointing out possible security aspects.

Characterization. To understand the characteristics of ASPP, previous work analyzed the view of ISPs [25], IXPs [10], and route collectors [11, 27, 68, 78]. Since their numbers were inconsistent, we refreshed and extended their findings by performing a 10-year analysis of the main properties of ASPP. In addition, our work is the first that focuses on prepending policies rather than only utilization rates.

Effectiveness. Swinnen et al. found—in simulations based on a degree-based network model—that ASPP cannot always move all traffic [65]. This finding was later confirmed in 2004 by Quoitin et al. when running measurements from a single vantage point connected to two upstreams [50] (similar to our effectiveness measurements). In contrast to their methodologies, we emulated and tested more than 100 real-world location combinations and showed that the effectiveness of ASPP varies substantially by location and the number of upstreams through which an AS announces the prepended prefix.

Security. Zhang et al. analyzed the potential of interception-attacks exploiting ASPP based on simulations on an AS Graph extracted from the public BGP data of RouteViews and RIPE RIS [77]. They show that well-connected ASes (e.g., Tier-1 ASes) are less prone to this type of attack and that longer prepends amplify their risks. We actively measure the security impact that ASPP has based on hijack emulations from various locations and experiments to identify ASes that remove prepends; we also observed that 18.6% of prepended prefixes have unnecessary prepend sizes that increase their exposure to attacks.

9 FINAL REMARKS

Despite mixed opinions about ASPP in the networking community, we find that ASPP is still very present on the Internet, and its utilization is slightly increasing. Surprised by this, we checked with operators and found that the main reasons are the simplicity of ASPP and the fact that it does not have any prerequisites. Our analysis of ASPP reveals that prepending policies are mostly stable over time; that ASes are using a wide range of policies when announcing their prefixes; and that prepend sizes are becoming polarized—with either one or more than three prepends.

We unexpectedly spot many ASes uniformly prepending (all) their prefixes to all neighbors, hence not influencing any remote routing decision. Via our conversations with operators, we identified poor housekeeping of BGP configurations, limited knowledge about BGP, and desire for stability as the possible leading causes. Our complementary analyses with traceroutes and cross-checks with CDN data confirm that, the limited visibility of public route collector projects cannot be the explanation for most of our observations.

During our interviews, many operators pointed out that using ASPP suffices to accomplish their ITE goals. Our active measurements confirm that ASPP is *effective*—since even small prepend sizes can steer the traffic of multiple routes—if used with many upstreams. When using only two upstreams, ASPP's effectiveness is dependent on the AS location.

We also discuss the security implications of ASPP. First, we show through active measurements that some ASes remove prepends, but it appears to be rare at the moment. Second, we find that ASPP can increase the spread of prefix hijacks, since the hijacked route is more attractive (than the actual route) to a larger fraction of ASes. Third, we detect that ASes originate 18% of the prepended prefixes with unnecessary prepends.

ASPP has value, and ASes are using it extensively on the Internet. However, as Internet paths are getting shorter (as the core is getting denser), the need for large prepend sizes is decreasing. Thus, given the security implications of large prepends and the fact that small prepends are often sufficient for moving traffic, we recommend network operators to review their prepending policies, removing unnecessary prepends and using small prepend sizes when performing ITE.

[15] We note that the fraction of prepended prefixes whose minimum number of prepends is zero that has ROAs is similar to the ones in the plot.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers and our shepherd, Rocky Chang, for their valuable feedback on our paper. We are also very thankful to all network operators for their valuable insights regarding the deployment, effectiveness, and security implications of ASPP. We are grateful to the three CDNs and the IXP who shared their data and the PEERING Testbed team for all their support during our active measurements. This work was supported by National Science Foundation grant CNS-1705024.

REFERENCES

[1] RIPE Routing Information Service. Available at http://www.ripe.net/ris/ Last accessed: May 30th, 2020.
[2] Routeviews Project – University of Oregon. Available at http://www.routeviews.org/ Last accessed: May 30th, 2020.
[3] AFRINIC. stats. Available at https://ftp.afrinic.net/pub/stats/afrinic/ Last accessed: May 31st, 2020.
[4] AMS-IX. AMS-IX breaks through 8 Tbps barrier, 2020. Available at https://www.ams-ix.net/ams/news/ams-ix-breaks-through-8-tbps-barrier Last accessed: April 14th, 2020.
[5] APNIC. stats. Available at https://ftp.apnic.net/apnic/stats/apnic/ Last accessed: May 31st, 2020.
[6] ARIN. stats. Available at https://ftp.arin.net/pub/stats/arin/ Last accessed: May 31st, 2020.
[7] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding Traceroute Anomalies with Paris Traceroute. In *Proceedings of the 2006 Internet Measurement Conference*, pages 153–158, 2006.

[8] H. Ballani, P. Francis, and X. Zhang. A Study of Prefix Hijacking and Interception in the Internet. *ACM SIGCOMM Computer Communication Review*, 37(4):265–276, 2007.

[9] T. Böttger, G. Antichi, E. L. Fernandes, R. di Lallo, M. Bruyere, S. Uhlig, and I. Castro. The Elusive Internet Flattening: 10 Years of IXP Growth. *CoRR*, 2018.

[10] S. H. B. Brito, M. A. Santos, R. dos Reis Fontes, D. A. L. Perez, and C. E. Rothenberg. Dissecting the Largest National Ecosystem of Public Internet eXchange Points in Brazil. In *International Conference on Passive and Active Network Measurement*, pages 333–345. Springer, 2016.

[11] A. Broido, E. Nemeth, and k. claffy. Internet Expansion, Refinement and Churn. *European Transactions on Telecommunications*, 13(1):33–51, 2002.

[12] CAIDA. Archipelago (Ark) Measurement Infrastructure. Available at https://www.caida.org/projects/ark/ Last accessed: June 2nd, 2020.

[13] CAIDA. The CAIDA UCSD AS Classification Dataset, 1st February 2020. Available at https://www.caida.org/data/as-classification Last accessed: May 30th, 2020.

[14] CAIDA. The CAIDA UCSD AS to Organization Mapping Dataset, May 4th, 2020. Available at https://www.caida.org/data/as_organizations.xml Last accessed: June 1st, 2020.

[15] R. K. Chang and M. Lo. Inbound Traffic Engineering for Multihomed ASs Using AS Path Prepending. *IEEE Network*, 19(2):18–25, 2005.

[16] K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao. Where the Sidewalk Ends: Extending the Internet AS Graph Using Traceroutes from P2P Users. In *Proceedings of the 5th International Conference on Emerging Networking Experiments And Technologies*, pages 217–228, 2009.

[17] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill. BGP hijacking classification. In *2019 Network Traffic Measurement and Analysis Conference*, pages 25–32. IEEE, 2019.

[18] T. Chung, E. Aben, T. Bruijnzeels, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, R. v. Rijswijk-Deij, J. Rula, et al. RPKI is Coming of Age: A Longitudinal Study of RPKI Deployment and Invalid Route Origins. In *Proceedings of the 2019 Internet Measurement Conference*, pages 406–419, 2019.

[19] Cisco. Influencing Inbound Path Selection by Modifying the AS_PATH Attribute, 2018. Available at https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-3se/3850/irg-xe-3se-3850-book/irg-prefix-filter.html Last accessed: April 14th, 2020.

[20] DE-CIX. Highest jump ever: DE-CIX Frankfurt reaches 9.1 Tbps, 2020. Available at https://www.de-cix.net/de/news-events/news/de-cix-frankfurt-reaches-9-1-tbps Last accessed: April 14th, 2020.

[21] DELL. set as-path, 2020. Available at https://www.dell.com/support/manuals/de/de/debsdt1/networking-z9100/z9100-on-9.14.2.6-cli-pub/set-as-path?guid=guid-f2652337-11a3-4dce-bc31-6bd3729bfbf3&lang=en-us Last accessed: April 4th, 2020.

[22] G. Di Battista, M. Patrignani, M. Pizzonia, and M. Rimondini. Towards Optimal Prepending for Incoming Traffic Engineering. In *3rd International Workshop on Internet Performance, Simulation, Monitoring, and Measurement (IPS MoMe 2005)*, 2005.

[23] B. Donnet and O. Bonaventure. On BGP Communities. *ACM SIGCOMM Computer Communication Review*, 38(2):55–59, 2008.

[24] R. Fanou, P. Francois, and E. Aben. On the Diversity of Interdomain Routing in Africa. In *International Conference on Passive and Active Network Measurement*, pages 41–54. Springer, 2015.

[25] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for Interdomain Traffic Engineering. *ACM SIGCOMM Computer Communication Review*, 33(5):19–30, 2003.

[26] FRRouting. Using AS Path in Route Map. Available at http://docs.frrouting.org/en/latest/bgp.html#bgp-router-configuration Last accessed: April 4th, 2020, 2020.

[27] J. Gamba, R. Fontugne, C. Pelsser, R. Bush, and E. Aben. BGP Table Fragmentation: what & who? In *Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*, 2017.

[28] Google. COVID-19 Community Mobility Reports, 2020. Available at https://www.google.com/covid19/mobility/ Last accessed: June 1st, 2020.

[29] E. Gregori, A. Improta, L. Lenzini, L. Rossi, and L. Sani. On the Incompleteness of the AS-level Graph: a Novel Methodology for BGP Route Collector Placement. In *Proceedings of the 2012 Internet Measurement Conference*, pages 253–264, 2012.

[30] A. Gupta, M. Calder, N. Feamster, M. Chetty, E. Calandro, and E. Katz-Bassett. Peering at the Internet's Frontier: A First Look at ISP Interconnectivity in Africa. In *International Conference on Passive and Active Network Measurement*, pages 204–213. Springer, 2014.

[31] C. Hadas Gold. Netflix and YouTube are slowing down in Europe to keep the internet from breaking, 2020. Available at https://edition.cnn.com/2020/03/19/tech/netflix-internet-overload-eu/index.html Last accessed: June 1st, 2020.

[32] R. Hiran, N. Carlsson, and P. Gill. Characterizing Large-scale Routing Anomalies: A Case Study of the China Telecom Incident. In *International Conference on Passive and Active Network Measurement*. Springer, 2013.

[33] IANA. Autonomous System (AS) Numbers. Available at https://www.iana.org/assignments/as-numbers/as-numbers.xhtml Last accessed: June 1st, 2020.

[34] Isolario Project. Available at https://isolario.it/ Last accessed: May 30th, 2020.

[35] Juniper. Example: Configuring a Routing Policy to Prepend the AS Path, 2019. Available at https://www.juniper.net/documentation/en_US/junos/topics/example/routing-policy-security-routing-policy-to-prepend-to-as-path-configuring.html Last accessed: April 4th, 2020.

[36] LACNIC. stats. Available at https://ftp.lacnic.net/pub/stats/lacnic/ Last accessed: May 31st, 2020.

[37] D. Madory. Excessive BGP AS Path Prepending is a Self-Inflicted Vulnerability, 2019. Available at https://ripe79.ripe.net/archives/video/187/ Last accessed: May 30th, 2020. We reference the questions after the talk.

[38] D. Madory. Excessive BGP AS Path Prepending is a Self-Inflicted Vulnerability, 2019. Available at https://blogs.oracle.com/internetintelligence/excessive-as-path-prepending-is-a-self-inflicted-vulnerability Last accessed: May 30th, 2020.

[39] P. Marcos. Can AS-PATH prepending compromise the security of Internet routing?, 2019. Available at https://blog.apnic.net/2019/06/18/can-as-path-prepending-compromise-the-security-of-internet-routing/ Last accessed: May 30th, 2020.

[40] A. Marder, M. Luckie, A. Dhamdhere, B. Huffaker, k. claffy, and J. M. Smith. Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale. In *Proceedings of the 2018 Internet Measurement Conference*, pages 56–69, 2018.

[41] D. McPherson and V. Gill. BGP MULTI_EXIT_DISC (MED) Considerations. RFC 4451, RFC Editor, March 2006.

[42] C. Michael Heusner. Marilia Mendonça breaks world record with live-streamed concert, 2020. Available at https://www.campaignlive.com/article/marilia-mendonca-breaks-world-record-live-streamed-concert/1680008 Last accessed: June 1st, 2020.

[43] Mikrotik. Main/Backup link setup, 2010. Available at https://wiki.mikrotik.com/wiki/Manual:Simple_BGP_Multihoming Last accessed: April 14th, 2020.

[44] MSK-IX. Internet traffic peak hit on March 30. =https://www.msk-ix.ru/en/press-center/news/?id=20200331, 2020. https://www.ams-ix.net/ams/news/ams-ix-breaks-through-8-tbps-barrier.

[45] NLnetLabs. Routinator. Available at https://github.com/NLnetLabs/routinator Last accessed: May 31st, 2020.

[46] W. Norton. 95th Percentile Internet Billing Method. Available at https://drpeering.net/white-papers/Ecosystems/95th-percentile-measurement-Internet-Transit.html Last accessed: June 1st, 2020.

[47] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. The (In)Completeness of the Observed Internet AS-level Structure. *IEEE/ACM Transactions on Networking*, 18(1):109–122, 2009.

[48] C. Pelsser, L. Cittadini, S. Vissicchio, and R. Bush. From Paris to Tokyo: On the Suitability of ping to Measure Latency. In *Proceedings of the 2013 Internet Measurement Conference*, pages 427–432, 2013.

[49] C. Petrie. BGP Even-More Specifics in 2017, 2017. Available at https://labs.ripe.net/Members/stephen_strowes/bgp-even-more-specifics-in-2017 Last accessed: April 14th, 2020.

[50] B. Quoitin, C. Pelsser, O. Bonaventure, and S. Uhlig. A performance evaluation of BGP-based traffic engineering. *International journal of network management*, 15(3):177–191, 2005.

[51] Rapid 7. Http get responses. Available at https://opendata.rapid7.com/sonar.http/.

[52] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, RFC Editor, January 2006. http://www.rfc-editor.org/rfc/rfc4271.txt.

[53] A. Reuter, R. Bush, I. Cunha, E. Katz-Bassett, T. C. Schmidt, and M. Wählisch. Towards a Rigorous Methodology for Measuring Adoption of RPKI Route Validation and Filtering. *ACM SIGCOMM Computer Communication Review*, 48(1):19–27, 2018.

[54] RIPE. stats. Available at https://ftp.ripe.net/pub/stats/ripencc/ Last accessed: May 31st, 2020.

[55] RIPE. 5.1 Allocations made by the RIPE NCC to LIRs, 2019. Available at https://www.ripe.net/publications/docs/ripe-733#51 Last accessed: April 14th, 2020.

[56] RIPE. The RIPE NCC has run out of IPv4 Addresses, 2019. Available at https://www.ripe.net/publications/news/about-ripe-ncc-and-ripe/the-ripe-ncc-has-run-out-of-ipv4-addresses Last accessed: June 1st, 2020.

[57] B. Schlinker, T. Arnold, I. Cunha, and E. Katz-Bassett. PEERING: Virtualizing BGP at the Edge for Research. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pages 51–67, 2019.

[58] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng. Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In *Proceedings of the 2017 Conference of the ACM Special Interest Group on Data Communication*, pages 418–431. ACM, 2017.

[59] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster, and E. Katz-Bassett. PEERING: An AS for Us. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, pages 1–7, 2014.

[60] Semaphore. 95th percentile bandwidth metering explained and analyzed. Available at https://www.semaphore.com/95th-percentile-bandwidth-metering-explained-and-analyzed/ Last accessed: June 1st, 2020.

[61] P. Sermpezis, V. Kotronis, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti. ARTEMIS: Neutralizing BGP Hijacking Within a Minute. *IEEE/ACM Transactions on Networking*, 26(6):2471–2486, 2018.
[62] K. Sriram, D. Montgomery, D. McPherson, E. Osterweil, and B. Dickson. Problem Definition and Classification of BGP Route Leaks. RFC 7908, RFC Editor, June 2016.
[63] F. Streibelt, F. Lichtblau, R. Beverly, A. Feldmann, C. Pelsser, G. Smaragdakis, and R. Bush. BGP Communities: Even more Worms in the Routing Can. In *Proceedings of the 2018 Internet Measurement Conference*, pages 279–292, 2018.
[64] T. Strickx. Technical Debt: an Anycast Story, 2018. Available at https://ripe77.ripe.net/archives/video/2222/ Last accessed: May 30th, 2020.
[65] L. Swinnen, S. Tandel, S. Uhlig, B. Quoitin, and O. Bonaventure. An Evaluation of BGP-based Traffic Engineering Techniques. Technical report, INFONET, October 2002.
[66] Team Cymru. The Bogon Reference. Available at https://team-cymru.com/community-services/bogon-reference/ Last accessed: May 31st, 2020.
[67] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark. To Filter or not to Filter: Measuring the Benefits of Registering in the RPKI Today. In *International Conference on Passive and Active Network Measurement*, pages 71–87. Springer, 2020.
[68] H. Wang, R. K. Chang, D. M. Chiu, and J. C. Lui. Characterizing the Performance and Stability Issues of the AS Path Prepending Method: Taxonomy, Measurement Study and Analysis. In *Proceedings of ACM SIGCOMM Asia Workshop*. Citeseer, 2005.
[69] Wikipedia. List of Internet exchange points. Available at https://en.wikipedia.org/wiki/List_of_Internet_exchange_points Last accessed: June 1st, 2020.
[70] Wikipedia. List of tier 1 networks. Available at https://en.wikipedia.org/wiki/Tier_1_network Last accessed: March 30th, 2020.
[71] F. Wohlfart, N. Chatzis, C. Dabanoglu, G. Carle, and W. Willinger. Leveraging Interconnections for Performance: The Serving Infrastructure of a Large CDN. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 206–220. ACM, 2018.
[72] World Health Organization. Coronavirus disease (COVID-19) advice for the public. Available at https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public Last accessed: June 1st, 2020.
[73] World Health Organization. Situation report - 132, 2020. Available at https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200531-covid-19-sitrep-132.pdf?sfvrsn=d9c2eaef_2 Last accessed: June 1st, 2020.
[74] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, et al. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *Proceedings of the 2017 Conference of the ACM Special Interest Group on Data Communication*, pages 432–445. ACM, 2017.
[75] B. Yeganeh, R. Durairajan, R. Rejaie, and W. Willinger. How Cloud Traffic Goes Hiding: A Study of Amazon's Peering Fabric. In *Proceedings of the 2019 Internet Measurement Conference*, pages 202–216, 2019.
[76] Y. Zhang. Method and System for Effective BGP AS-Path Pre-pending, May 23 2013. US Patent App. 13/300,372.
[77] Y. Zhang and M. Pourzandi. Studying Impacts of Prefix Interception Attack by Exploring BGP AS-PATH Prepending. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pages 667–677. IEEE, 2012.
[78] Y. Zhang and M. Tatipamula. Characterization and Design of Effective BGP AS-PATH Prepending. In *2011 19th IEEE International Conference on Network Protocols*, pages 59–68. IEEE, 2011.
[79] E. Zmijewski. Longer is not always better, 2009. Available at https://dyn.com/blog/longer-is-not-better/ Last accessed: June 1st, 2020.
[80] E. Zmijewski. Reckless Driving on the Internet, 2009. Available at https://dyn.com/blog/the-flap-heard-around-the-world/ Last accessed: June 1st, 2020.

A ETHICAL CONSIDERATIONS

To conduct our study, we relied on active as well as passive measurements. When we used the PEERING testbed to actively announce prefixes to the Internet, we ensured that we did not overwhelm any networks by waiting 15 minutes between consecutive announcements. When actively sending traffic from the PEERING testbed, we ensured not to cause any harm through the following mechanisms: *(i)* we sent probing packets at a low rate, i.e., each target IP was probed with one ICMP, one TCP, and one UDP probe once per minute. *(ii)* we avoided traffic bursts by spreading the sending of probes equally throughout a one-minute interval. *(iii)* we included our contact info in the payload of each probe providing details on how to opt-out of the probing process. We have not received complaints nor requests to opt-out of the experiments during the entire duration of our active experiments.

While most of our passive datasets are publicly available, we cannot share any of the data received from CDNs and the European IXP for validation purposes. As this limits the possibility for others to take action based on our results, we tried, whenever possible, to reach the network operators of ASes that consistently announce uniform prefixes.

B MONITOR FILTERING

Figure 18 shows the distribution of the fraction of monitors that observe a given prefix based on all routing information available on January 15th of each year. While prefix visibility has increased over the decade, we find a clear separation between two distinct regions in the plot regardless of the year. On the leftmost side, we have locally visible prefixes (seen by less than 20% of monitors), and on the rightmost side, globally visible prefixes (seen by over 80% of monitors). Based on this finding, we decided to remove all routes to prefixes observed by less than one-third of all monitors, as indicated by the threshold line. Notably, picking any other threshold between 0.2 and 0.8 only results in negligible differences.

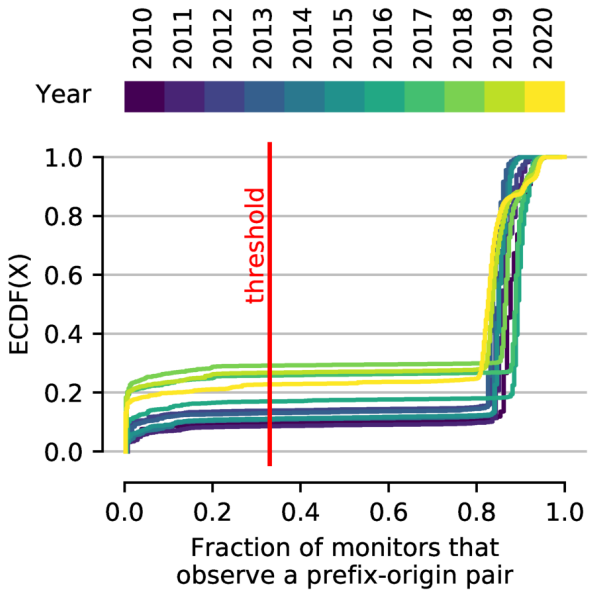


Figure 18: Fraction of monitors that observe each prefix-origin pair.

C EFFECTIVENESS TIMELINE

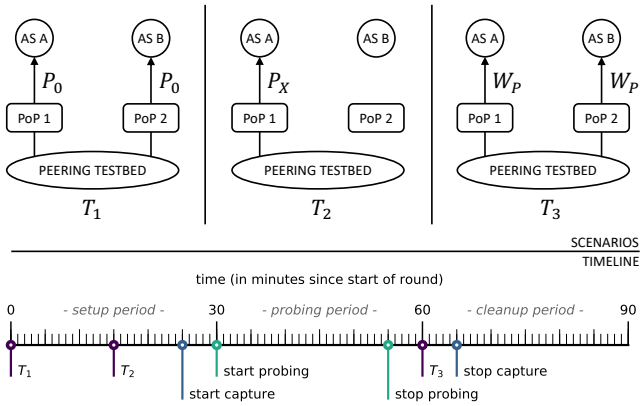


Figure 19: Effectiveness experiment: Timeline and experimental setting (P_X: announcement of prefix P with $X \in 0, \ldots, 3$ prepends; W_P: withdraw P.

In Figure 19, we depict the timeline of events and the configuration scenarios from each iteration of our effectiveness experiment. First, see Scenario T_1, we create a baseline by announcing our prefix P via all upstreams without any prepending. After waiting 15 minutes to allow BGP to converge, we announce P with X prepends via the chosen upstreams, see Scenario T_2. After again waiting for 15 minutes to allow BGP to converge, we conclude the *setup period* and start a 25-minute *probing period*. Each probe consists of ICMP, TCP, and UDP pings triggered once per minute to all targets. To reduce probing bursts, we spread the packets evenly across the one-minute time interval. Before the cleanup, we have a 5-minute break to ensure that the last responses can arrive before we withdraw the prefix. After the break, we start the *cleanup period* with the withdrawal of the announcements in every upstream, see Scenario T_3. To allow for BGP to converge and to minimize the risk of BGP Route Flap Damping, we wait for 30 minutes before starting a new iteration.